

TRANSMITTAL

Transmitted to: Dr. Brewster Kahle
WAIS Inc.
1040 Neal Drive
Menlo Park, CA 94025

Journal: Journal of the American Society for Information Science
Article: 495

Project number: 1137
Date: 5-25-93

Herewith:

Manuscript ☒ Pages To correspond
Galley proofs ☐ set(s) _____
Foul galleys ☐ _____
Page proofs ☒ 1 set(s) pp. 1-15
Foul page proofs ☐ _____
Final page proofs ☐ set(s) _____
Repro/Mechanicals ☐ set(s) _____
Art proofs ☐ set(s) _____
Foul art proofs ☐ _____
Revised art proofs ☐ set(s) _____
Original art/photos ☐ _____
Halftone negatives ☐ _____
Disk/Tape ☐ _____
Note ☐ _____

fax: 212-850 6052
Elihu Solandz
John Wiley

Transmitted by: Miriam Brady

Ship via: First Class

For _____

Date _____ Time _____

WHILE YOU WERE OUT

M William Solondz

From John Wiley

Phone No. 212-750-0183

Area Code Number Extension

TELEPHONED	<input type="checkbox"/>	URGENT	<input type="checkbox"/>
PLEASE CALL	<input type="checkbox"/>	WANTS TO SEE YOU	<input type="checkbox"/>
WILL CALL AGAIN	<input type="checkbox"/>	CAME TO SEE YOU	<input type="checkbox"/>

RETURNED YOUR CALL ☐

Message RE: Our article

Operator

Lillian Solondz
Editorial Production
JOURNALS DEPT.
PROFESSIONAL, REFERENCE, &
TRADE GROUP
JOHN WILEY & SONS, INC.
605 THIRD AVENUE
New York, New York 10158
USA

WILEY

REPRINT BILLING DEPARTMENT • 605 THIRD AVENUE • NEW YORK, NY 10158-0012
PHONE: (212) 850-8776; FAX: (212) 850-8888

PREPUBLICATION REPRINT ORDER FORM

Please complete this form even if you are not ordering reprints. This form **MUST** be returned with your corrected proofs and original manuscript. Your reprints will be shipped approximately 4 to 6 weeks after publication. Reprints ordered after printing are substantially more expensive.

JOURNAL : JRNL OF AM. SOC. FOR INFO SCIENCE VOLUME ISS UE
TITLE OF MANUSCRIPT Interfaces for Distributed Systems of Information Servers
MS. NO. 495 NO. OF PAGES AUTHOR(S) B. Kahle, H. Morris

REPRINTS 8 1/4 X 11					
No. of Pages	100 Reprints	200 Reprints	300 Reprints	400 Reprints	500 Reprints
	\$	\$	\$	\$	\$
1-4	181	336	468	596	705
5-8	256	472	663	840	995
9-12	319	590	828	1,052	1,245
13-16	383	708	996	1,264	1,495
17-20	426	790	1,107	1,404	1,665
21-24	488	902	1,269	1,608	1,900
25-28	539	998	1,404	1,780	2,105
29-32	594	1,098	1,542	1,956	2,315
33-36	653	1,206	1,695	2,152	2,545
37-40	712	1,316	1,851	2,348	2,775

**** REPRINTS ARE ONLY AVAILABLE IN LOTS OF 100. IF YOU WISH TO ORDER MORE THAN 500 REPRINTS, PLEASE CONTACT OUR REPRINTS DEPARTMENT AT (212)850-8776 FOR A PRICE QUOTE.**

COVERS		
100 Covers - \$90	•	200 Cover - \$145
400 Covers - \$255	•	500 Covrs - \$325
	•	300 Covers - \$200
		Additional 100s - \$65

☐ Please send me _____ reprints of the above article at..... \$ _____
☐ Please send me _____ covers of the above journal at..... \$ _____
Please add appropriate State and Local Tax (Tax Exempt No. _____) \$ _____
Please add 5% Postage and Handling..... \$ _____
TOTAL AMOUNT OF ORDER** \$ _____

****International orders must be paid in U.S. currency and drawn on a U.S. bank**

Please check one: ☐ Check enclosed ☐ Bill me ☐ Credit Card
If credit card order, charge to: ☐ American Express ☐ Visa ☐ MasterCard
Credit Card No. _____ Signature _____ Exp. Date _____

Bill To:
Name _____
Institution/Company _____
Department _____
Address _____
City _____ State _____ Zip _____
Purchase Order No. _____

Ship To:
Name _____
Address _____
City _____ State _____ Zip _____

Received 4-22-93

pg. 1-15

BG# 1137

15 days 5-12-93

18 days 5-17-93

Journal of the American Society of Information Science

published by
JOURNALS DEPT.
SCI/TECH DIVISION
JOHN WILEY & SONS, INC.

44(8)

Production No.: JA 495 CTF ENCL. _____ To Come X

Author(s): Brewster Kahle and Harry Morris

Title: Interfaces for Distributed Systems of
Information Servers

Address of
Corresponding Author: Brewster Kahle

Mendo Park, CA 94025

Tel. # _____

No. of Pages: 35 No. of Figures: 16 No. of Tables: 0

EST H PP: 18

art due date 5-5

Proofs to:

1 set marked proof to Publisher

1 set of proof to editor, Prof. Don Kraft (label enclosed)

1 set marked proof, ms., artwork, 1 "Notice to Authors and Reprint Order Form" to author (label enclosed)

Author: _____

Running Head: _____

Received Date: X Revised Date: X

Wiley 4/12/93



WILEY

Publishers Since 1807

John Wiley & Sons, Inc.
605 Third Avenue
New York, NY 10158
(212) 850-6000
Telex 12-7063
Cable JONWILE

JOURNAL: *JASIS*

PRODUCTION NUMBER: *JA-495*

Dear Author:

Enclosed please find your original manuscript with copyeditor's corrections and one set of page proofs. Proofread the proofs against the manuscript, using a red pen to mark changes directly on the proofs. Double-check figures and tables and be sure to respond to queries that appear on the pages listed below. Finally, please return original manuscript, corrected proofs, tables, and figures to Wiley as soon as possible.

Authors outside the U.S.: Your manuscript and proofs have been sent to you via the quickest manner possible. Please assist us in keeping your article on schedule by returning the manuscript, proofs, and figures to us as quickly as possible. Your cooperation is greatly appreciated.

Disk Submission: If you have submitted your manuscript on computer disk, you still need to proofread the page proofs carefully against the original manuscript.

Thank You.

AUTHOR: Please answer the queries on the following pages:

17, 19, 20, 22, 34

Please sign the attached copyright transfer & return it with your proofs. Thanks.

Au: Please cite Figures 1 through 11 in text.

JA 495

Copyright Transfer Agreement

Under U.S. copyright law the transfer of copyright from the author needs to be explicitly stated in writing to enable the publisher to publish and disseminate the author's work to the fullest extent. Therefore, this Agreement must be signed and returned to us before we can process your manuscript.

Copyright Transfer Agreement

The undersigned author has submitted a manuscript entitled _____

_____ (the "Work")

for publication in *JASIS: Journal of the American Society for Information Science*, (the "Journal") published by John Wiley & Sons, Inc.

A. The author transfers to John Wiley & Sons, Inc. (the "Publisher") during the full term of copyright, the exclusive rights comprised in the copyright of the Work, including but not limited to the right to publish the Work and the material contained therein throughout the world, in all languages, and in all media of expression now known or later developed, and to license or permit others to do so.

B. Notwithstanding the above, the author retains the following:

1. Proprietary rights other than copyright, such as patent rights.
2. The right to make copies of all or part of the Work for the author's use in classroom teaching.
3. The right to use, *after publication*, all or part of the Work in a book by the author, or a collection of the author's work.
4. The right to make copies of the Work for internal distribution within the institution which employs the author.
5. The right to use figures and tables from the Work, and up to 250 words of text, for any purpose.
6. The right to make oral presentations of material from the Work.

The author agrees that all copies made under any of the above conditions will include notice of copyright and a citation to the Journal.

C. In the case of a Work prepared under U.S. Government contract, the U.S. Government may reproduce, royalty-free, all or portions of the Work and may authorize others to do so, for official U.S. government purposes only, if the U.S. Government contract so requires. A copy of the contract must be attached.

D. If the Work was written as a work made for hire in the course of employment, the Work is owned by the company/employer which must sign this Agreement in the space provided below. In such case, the Publisher hereby licenses back to such employer the right to use the Work internally or for promotional purposes only.

E. The author represents that the Work is the author's original work. If the Work was prepared jointly, the author agrees to inform the co-authors of the terms of this Agreement and to obtain their permission to sign on their behalf. The Work is submitted only to this Journal, and has not been published before. (If excerpts from copyrighted works are included, the author will obtain written permission from the copyright owners and show credit to the sources in the Work.) The author also represents that, to the best of his or her knowledge, the Work contains no libelous or unlawful statements, does not infringe on the rights of others, or contain material or instructions that might cause harm or injury.

Author's signature and date

Check one:

- ☐ Author's own work
☐ U.S. Government work
☐ Work made for hire for Employer

Typed or printed name

Institution or company (Employer)

Note to U.S. Government Employees

A Work prepared by a U.S. federal government employee as part of his/her official duties is called a "U.S. Government work," and is in the public domain in the United States; in such case, Paragraph A above applies only outside the United States. Please attach a copy of any applicable policy of the author's agency.

If the Work was prepared jointly, and any co-author is not a U.S. Government employee, it is not a U.S. Government work. That co-author should be delegated by the other co-authors to sign this Agreement. If the Work was not prepared as part of the employee's duties, it is not a U.S. Government work.

**WILEY**

Publishers Since 1807

John Wiley & Sons, Inc.605 Third Avenue
New York, NY 10158
(212) 850-6000

Notice to Authors

Enclosed please find the manuscript and proofs for your article. No further proofs will be sent to you.

Please return the corrected proofs within 48 hours of receipt to ensure prompt publication. If your proofs are not returned within this period, we reserve the right to publish your paper as typeset with no further corrections. The author is solely responsible for any errors that are undetected by his proofreading.

You alone are responsible for checking the proofs of your article and for making all necessary corrections on the proofs. Please be sure to answer all queries posed in the margins of the manuscript and proof. We reserve the right to charge authors for any alterations to the text or illustrations which are not due to typesetting errors, at a rate of no less than \$2.00 per affected line. No further changes may be made once the marked proofs have been returned to us for processing.

Unless otherwise indicated, return the enclosed material (manuscript, illustrations, marked proofs) within 48 hours by first class mail (air mail if sent from outside the United States) to: **Lillian Solondz**

*Express mail would certainly be appreciated. Or you could fax your corrections or call them in.

Fax: (212) 850-6052

Tel: (212) 850-6183

Journals Editorial/Production
Professional, Reference, and Trade Group
John Wiley & Sons, Inc.
605 Third Avenue
New York, N.Y. 10158
U.S.A.

To order reprints of your article, complete the order form enclosed. The reprint order form **MUST** be returned with your manuscript. Payment may be made by personal check, credit card, or a purchase order from your institution. Make sure a copy of this form is returned with your purchase order or payment. International orders must be paid in U.S. currency and drawn on a U.S. bank. Reprints should be ordered prior to publication since postpublication orders are subject to a back-to-press surcharge of \$55.00 per article.

Proofreader's Marks

Mark in Margin	Mark in Proof	Meaning	Mark in Margin	Mark in Proof	Meaning
cap/	are capitalized in English titles	Capital letter	eq #/	information, as / when, / for/ex-	Equalize spacing
lc/	only the first word and proper	Lower-case letter	fl. l./	Example, the volume and page of	Move left
tr/	nouns in French and Italian	transpose	et al/	a periodical, cited and the	Let type stand
s/	titles and all nouns in German.	Insert comma	w. b. year/	added for the convenience	Wrong type font
e/	Setting the volume number of	Delete and close up	R/	of the reader. The following	Begin new paragraph
X/	a periodical in distinctive type	Fix broken letter	⊙/	illustrates a bibliographic item	Change period to colon
I/	(italics or bold face) and page	Delete hyphen; leave space	D. C. /	GAY, CHARLES M. and Charles	Set in small caps
cut type/	numbers in ordinary type makes	Cut type	ital/	DE VAN FAWCETT, Mechanical and	Set in italics
v/	"Vol" and "p" unnecessary	Insert quotes	for/	Electrical Equipment & Buildings	Change of to for
x/	The year of publication is en	Insert hyphen	rom/	(John) Wiley & Sons, New York	Set in roman
e/	closed in parentheses only when	Substitute e for i	o/#/	second edition, p 229, 1945	Close up, insert space
cut line/	it is actually parenthetical	Shadow line			

YOU CAN HELP US ACHIEVE THE LARGEST POSSIBLE READERSHIP FOR YOUR ARTICLE

Please take a moment to give us the names and addresses of any organizations or companies that might be interested in purchasing reprints of your article, and return this form along with your page proofs. If you prefer to send it separately, please mail it to:

John Wiley & Sons, Inc.

605 Third Avenue

New York, NY 10158

Attn: Craig Woods

Reprint Department, 9th Floor

Your Article Number 495

Organization _____

Address _____

Contact Name _____

Phone _____

Fax _____

Organization _____

Address _____

Contact Name _____

Phone _____

Fax _____

JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE RUNNING FEET SHEET

PROD. NO. JA-495

© 1993 by John Wiley & Sons, Inc.

JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE ⁴⁴~~43~~():00-00, 1993

CCC 0002-8231/93/000000-00

Running Feet for internal pages

Even pages:

LRF 00 JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE--Month 1993

Odd pages:

RRF 00 JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE--Month 1993

OTA TO COM

JA 495

Wiley

TI

Interfaces for Distributed Systems of Information Servers

NI

LI

Brewster Kahle and Harry Morris

WAIS Inc.

1040 Noel Drive, Menlo Park, CA 94025

brewster@wais.com, morris@wais.com

NI

LI

Jonathan Goldman

Thinking Machines Corporation

1010 El Camino Real, Menlo Park, CA 94025

jonathan@think.com

NI

LI

Thomas Erickson

Apple Computer

20525 Mariani Avenue, MS 76-3H, Cupertino, CA 95014

thomas@apple.com

NI

LI

John Curran

NEARnet

10 Moulton Street, Cambridge, MA 02138

jcurran@nic.near.net

ABSTRACT

A1 Interfaces for information access and retrieval are a long way from the ideal of the electronic book that you can cuddle up with in bed. Nevertheless, today's interfaces are coming closer to supporting browsing, selection, and retrieval of remote information by non/technical users.

This ^{article} paper describes five interfaces to distributed systems of servers that have been designed and implemented: WAISStation for the Macintosh, XWAIS for X-Windows, GWAIS for Gnu-Emacs, SWAIS for dumb terminals, and Rosebud for the Macintosh. These interfaces talk to one of two server systems: the Wide Area Information Server (WAIS) system on the internet, and the Rosebud Server System, on an internal network at Apple Computer. Both server systems are built on Z39.50, a standard protocol, and thus support access to a wide range of remote databases.

The interfaces described here reflect a variety of design constraints. Such constraints range from the mundane ^m-coping with dumb terminals and limited screen space ^m-to the challenging. Among the challenges addressed are how to provide passive alerts, how to make information easily scannable, and how to support retrieval and browsing by nontechnical users. There are a variety of other issues which have received little or no attention, including budgeting money for access to "for pay" databases, privacy, and how to assist users in finding out which of a large (changing) set of databases holds relevant information. We hope that the challenges we have identified, as well as the existence and public availability of source code for the WAIS system, will serve as a stimulus for further design work on interfaces for information retrieval.

H1

INTRODUCTION

B1

It requires little prescience to predict that one day computers will put an ocean of information at the finger tips of a vast population of users. However, although there is a considerable amount of information available from remote sources, the bulk of it is accessible only to information professionals, or users with technical backgrounds. A variety of obstacles effectively block the ordinary user from accessing information via the computer. These obstacles include the difficulty of locating appropriate information sources, the cumbersome maneuvers needed to get online and to connect to remote sources, and cryptic query languages. Furthermore, even if a user has succeeded in accessing a remote information source, it is likely that it will have its own special purpose interface, which may or may not support the user's needs.

In this ^{article} paper we describe two systems: ^{Wide Area Information Servers (WAIS)} and ^{Rosebud} which provide a protocol-based mechanism for accessing a variety of remote, full-text information servers. These systems have the potential for supporting a single interface to a wide variety of information sources, and offer a good platform on which to explore the design of interfaces for information retrieval. After a summary of existing information retrieval systems, we describe the server systems, and then describe the five interfaces to them. In the course of these descriptions we discuss design constraints, interface issues, and practical matters which had an impact on the designs. We conclude with a summary, and some remarks on important issues that have not been addressed, and an invitation for other investigators to use the WAIS system as a platform for exploring interfaces to multiple, remote information sources.

H1

BACKGROUND

H2

Existing Systems

While a review of all existing systems is beyond the scope of this ^{article} paper, it is useful to list a number of the most popular or significant interfaces for information retrieval.

Commercial interfaces for accessing ^{fulltext} resources on computers can be broken down into dialup services, local file access, and LAN-based access tools. Dialup systems such as Dialog and Dow Jones offer TTY interfaces to users, with menus and command lines being the dominant access tools. Some dialup services are offering client programs that run on personal computers to add graphical interfaces such as "Navigator" by Compuserve. In general, these interfaces are unique to the information provider. Local file access through ^{fulltext} indexing has been achieved in command line form (e.g., the Unix command "grep") and in screen-based interfaces (e.g., ON Location [ON], and Digital Librarian [NeXT]). These interfaces often give browsing

and searching capabilities for local files. Some of these interfaces have been stretched to work with files on file servers. LAN-based access tools usually use some sort of query language to access servers on the net, such as Verity's Topic system (VERITY), and numerous library systems. These query languages require some user training. Integrated tools for cross-platform, cross-vendor information access are not currently available in other systems.

A variety of research projects have explored information retrieval systems. The SuperBook project (Egan, 1989) targets users of static information. Project Mercury (Ginther-Webster, 1990) is a remote library searching system that uses a client-server model. Information Lens (Malone, 1986) is a structured email system for assisting in managing corporate information. NetLib for software (Dongarra, 1987) and Mosis for information on how to fabricate chips (Mosis) are examples of email-based information retrieval systems.

H2) The WAIS and Rosebud Projects

The two systems of information servers described in this ^{article} grew out of two, partially entwined projects: WAIS, and Rosebud. A goal of both projects was to define an open protocol that would allow any user interface or information server that talked to the protocol to interact with any other component that used the protocol. From the user's perspective, this would mean that user interfaces and information sources could be mixed and matched, according to the user's needs.

WAIS started as a joint project between Thinking Machines Corporation, Apple Computer, Dow Jones & Co., and KPMG Peat Marwick (Kahle, ^{Medlar} et al, 1991a). The proximate goal was to define the open protocol and demonstrate its feasibility by implementing and demonstrating a multi-vendor system which provided ordinary users with access to a variety of remote databases. Thinking Machines contributed its Connection Machine-based retrieval technology, Apple contributed its expertise in user studies and interface design, and Dow Jones & Co. provided access to its commercial information sources. KPMG Peat Marwick provided access to its corporate data, and served as a site for user studies and testing. The WAIS system was installed at KPMG Peat Marwick and enabled the designers to study the success of the system in a real world context. The WAIS system uses pseudo-natural language queries, relevance feedback to refine queries, and accesses fulltext, unstructured information sources. These technologies were used because they had already been tested independently, thereby leading to faster implementation of the complete system. The WAIS system will be described in more detail in the next section.

During the same period, the Rosebud project was underway within Apple. Rosebud's goal was to serve as an internal platform for research into system architecture and human interface issues, and as a consequence, employed a variety of more experimental technologies and was tested in-house. Like WAIS, Rosebud was

based on user studies conducted at KPMG Peat Marwick, and used the same underlying protocol, Z39.50. The details of the Rosebud Server System will be described in a different paper. *study*

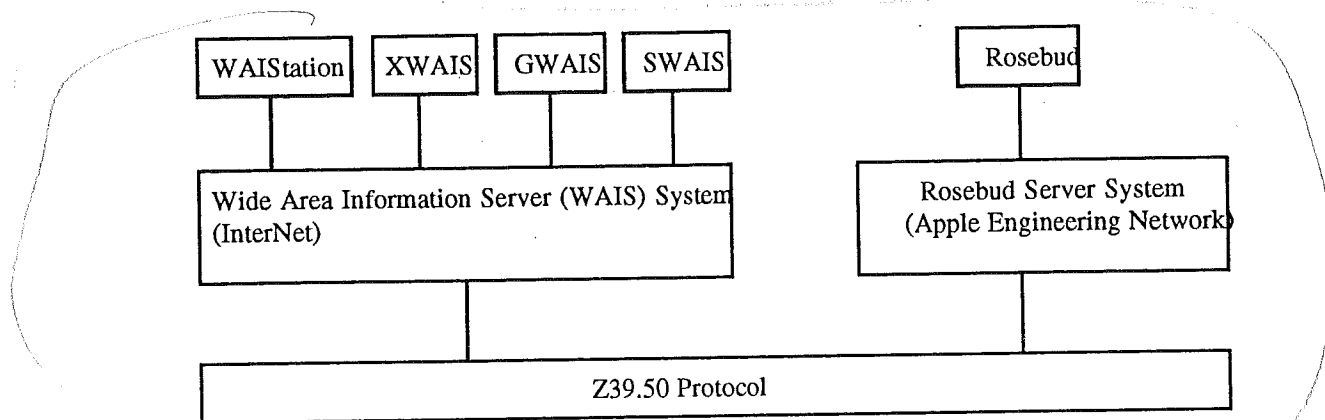


Figure 1

The interfaces to the WAIS and Rosebud server systems, and the protocol.

FIG 1

After the collaborative phase of the WAIS project came the Internet experiment. In this phase of WAIS, source code for the open protocol, information servers, and for several interfaces were made freely available over the Internet. In addition, Thinking Machines established and maintained a directory of information servers, that WAIS users could query to find out about available information sources. This phase of WAIS is still in progress, and has resulted in the creation of new interfaces, the availability over the Internet of more than a hundred servers on three continents, and over 100,000 searches of the directory of servers. In the first six months of the Internet experiment, approximately 4000 users from 20 countries have tried this system, with no training other than documentation (Kahle, ^{Goldman, Morris & Shen} et al, 1991b). Administrators of popular information servers indicate that they are getting over 50 accesses a day from many countries.

The WAIS System

WAIS employs a client-server model using a standard protocol (based on Z39.50) to allow users to find and retrieve information from a large number of servers. The client program is the user interface, the server does the indexing and retrieval of documents, and the protocol is used to transmit the queries and responses. Any client which is capable of translating a user's request into the standard protocol can be used in the system. Likewise, any server capable of answering a request encoded in the protocol can be used.

A WAIS server can be located anywhere that one's workstation has access: on the local machine, on a network, or on the other end of a modem. The user's workstation keeps track of a variety of information about

each server. The public information about a server includes how to contact it, a description of the contents, and the access cost.

4 The WAIS protocol (Davis, et al, 1990) is an extension of the existing Z39.50 standard (NISO, 1988) from NISO. It has been augmented where necessary to incorporate many of the needs of a full-text information retrieval system. To allow future flexibility, the standard does not restrict the query language or the data format of the information to be retrieved. Nonetheless, a query convention has been established for the existing servers and clients. The resulting WAIS Protocol is general enough to be implemented on a variety of communications systems.

9 The WAIS clients will be described in detail in the next several sections. However, all of them work in a basically similar way. On the client side, queries are expressed as strings of words, often pseudo-natural language questions. The client application then packages the query in the WAIS protocol and transmits it over a network to one or more servers. The servers receive the transmission, translate the received packet into their own query languages, and search for documents satisfying the query. The lists of relevant documents are then encoded in the protocol and transmitted back to the client. The client decodes the response and displays the results. The documents can then be retrieved from the server. The documents can be in any format that the client can display such as word processor files or pictures.



H2 WAISTATION: AN INTERACTIVE QUERY INTERFACE

WAISTation At A Glance

Target Machine	Macintosh Plus and above, 9" Monochrome screen.
Effort	1 person-year
Number of Users	2000
Status	Finished, freely distributed
Language	ThinkC
Communications	TCP/IP and Modem (not supported)
Designer	Harry Morris
Organization	Thinking Machines
Availability	Available for anonymous FTP from /public/wais/WAISTation*.sit.hqx@think.com
Design Goals	Implementable quickly, support interactive queries well, changeable based on user's comments, make something very simple to learn (partner friendly). Try out many ideas: interactive queries, passive alerting, asking multiple servers.
Used	In a study with accountants and tax consultants at KPMG: very good user acceptance. In the Internet experiment: estimated that half of the uses of WAIS are using WAISTation. (Based on when the directory of servers did not work for Macintoshes, usage dropped to half).
Problems	Dealing with the directory of servers (s). Modem code was difficult to get right.

Note
Author asked to provide this single spaced in paragraphs

WAISTation was designed for use in the WAIS experiment at KPMG Peat Marwick. As such, we needed an interface that would be easy to use, and would encourage successful searches by users untrained in search techniques. Peat Marwick often sends its employees into the field toting their Macintosh SE's along for use as portable computers. Thus we had to design the interface to run on a ^{nine} 9-inch black-and-white screen, and make minimal demands on CPU and memory. Furthermore, WAISTation was designed for use over modems and slow LANs.

112

Design Rationale

¶ In designing WAISStation, we were informed by two metaphors: search as conversation, and storage by file folder. The process of formulating an effective search is highly interactive. Of the documents which match a query, the ones which match "best" are displayed. One or more may be of interest, in which case, they can be fed back to the system, interactively improving the search. We choose to view this process as a conversation. Thus the initial natural language question becomes that starting point for give and take between the user and the server(s). Relevance feedback provides the context for the question. As the search proceeds, some results may suggest alternative searches or branches of the conversation. This is provided for by allowing several questions to evolve at the same time.

¶ Eventually one or more questions may be refined to the point where they are finding consistently good results. At this point, the question can be automated, becoming a dynamically updated file folder. At intervals these questions wake up and query their servers. The results are stored in the results field for later inspection. They can be thought of as regular Macintosh folders, except as augmented with a charter describing how to keep their contents up to date.

¶ This parallel with the Macintosh folder structure suggested a drag and drop construction for the user interface itself. Constructing a question is a three-step process: typing the key words, specifying the servers to use, and specifying the relevant documents to feed back. If we think of questions like Macintosh folders, we can use the Macintosh's drag-and-drop mechanism for putting sources and relevant documents into a question. This approach makes WAISStation's mechanics instantly familiar to users of the Macintosh finder.

112

Human Interface

¶ When WAISStation starts up, two windows appear: one contains the users available Sources (see below) and one contains the users saved Questions. Sources are identified by an eye icon, questions by a question mark icon.

¶ Double clicking on a question icon opens the stored question, including any new results found since the last time it was examined. The top half of the question window contains a field in which to type key words (the natural language part of the question), a list of relevant documents, a list of sources, and a list of result headlines. Sources can be added to the question by selecting a source icon (in the Sources window), and dragging it into the question. Relevant documents are specified in the same way.

¶ Result documents, returned by the servers, can be examined by double clicking on their icon. Note that the result list contains a graphical indication of how well each document matches the query. The original graphic was a series of 0 to 4 stars, similar to the ratings found in TV Guide. We thought that this rating scheme would be easily recognized. Experience proved that the stars did not provide enough information to be

recognized or to discriminate among the documents. Latter versions of the software replaced the stars with a horizontal bar giving 20 levels of resolution.

Any of the resulting documents can be opened and viewed in its own window. WAISStation supports plain ASCII documents as well as PICT format pictures. Text windows automatically scroll to the position which the server considers the most relevant part of the document. This allows the user to quickly determine if a file is useful. In order to perform well over slow communications channels (modems and slow LANs) the text is downloaded on demand in 15 line chunks. The keywords used in the query are automatically highlighted in bold.

Sources are specially formatted text files which describe information servers and how to get to them. Double clicking on a source displays a window with several controls. The top part is information specified by the server itself:

- a pop-up menu to specify the method of contacting the server (ip-address/tcp-port, modem number and speed, or location of a local index);
- a script to run after logging in (for use by modems);
- a database to search (servers can support multiple databases);
- a display of when the server is updated, how much it costs to search; and
- a textual description of the databases' contents.

The bottom half of the source window allows the user to specify personal information about the server:

- when to contact it (for automatic update);
- when it was last contacted;
- how much to spend on it;
- how much credence its results should be given (this is used to scale document scores, which helps in the sorting of responses to questions asked of multiple servers);
- the number of documents to ask for when searching it; and
- the font and type size to use when displaying plain text results (important to publishers).

Several of these fields are merely place holders in the current implementation. In particular, budget and confidence have not been implemented yet since there are no for-pay servers yet, and the number of sources is still relatively small.

- ④ Source files can also be retrieved from servers. This allows users to search servers whose database elements are pointers to other servers. The results can be used as targets for further searches. An experimental directory of servers is being maintained on the Internet.

1/2 Implementation

- ④ WAISStation was implemented in Think C 4.0 using the object oriented class library. It took about a man-year of effort. The most difficult parts were the automatic update facility and the communications. Automatic Update required the ability to do background processing - which is not a normal part of the Macintosh operating system. Communications were difficult primarily because we were simultaneously debugging the Z39.50 protocol, modem code, and the (then new) Apple Communications Toolbox. We eventually left modems unsupported, and replaced the Communications Toolbox with direct calls to MacTCP. Through this experience we found that communications speeds of less than 9600 baud were barely tolerable for interactive text retrieval.

1/2 Observations

- ④ We estimate that WAISStation is now in use by over 2000 users in 20 countries. The common user complaints center around configuring MacTCP, using (the undocumented) directory-of-servers, and avoiding a bug requiring the software to be installed on the start-up disk.

- ④ We have noticed several shortcomings in the current design:

- B3
- ④ • Users want access to their own data. WAISStation is capable of searching a Macintosh-based inverted index file, but we unbundled the index builder when we realized how much work it would take to make it useful under Macintosh OS. OnLocation (On Technology) is an implementation of a Macintosh indexer that could be used.
 - ④ • Interaction with the directory of servers is incomplete. It is not obvious which search results are source files, and what to do with the ones that are. It should be possible to drag a retrieved source directly into a question's source window, but the present interface requires that it be saved first. The lesson we learned was that special cases should be handled specially, rather than forcing users to use general techniques "for consistency's sake".
 - ④ • Printing documents and searching for keywords in documents (find/find-next) are simple functions which users expect.

B3

- People want to see their documents in their original form. WAStation currently only displays ASCII and PICT. This can be fixed with format filters such as Claris' XTND, at the expense of the ability to download arbitrary sections of a document, since such filters require that the document be processed from the beginning.
- Relevance feedback was not obvious. Users unfamiliar with the use of relevance feedback did not think to use it ^{it} it needs to be made more automatic. One way to do this might be to extend the notion that a question is a conversation, with relevance feedback as context (or body language) ^{clients or servers} clients or servers can be written that watch their users, and deduce which documents were relevant based on which ones were read. A simpler approach might be to always do relevance feedback, presenting the results in a "see also" list. We tried this, but the Macintosh was too slow to make it useful.
- Communications over 2400-baud modems are too slow to support interactive queries. We found that 9600 baud is barely acceptable, while 56Kb is sufficient to support several users.
- The finder-like interface (drag and drop) is not obvious. Even though the Macintosh Finder is based on drag and drop, no one expected it in an application. Once users were shown what to do, it became natural. It was also not necessarily the best use of screen space, since it required that both the start and end of the drag be visible on the screen at the same time. Another anomaly worth mentioning is the fact that although we were simulating the finder, we had no "trash can" analogy. Removing a source was accomplished by dragging it onto the desk top and dropping it there, which confused some users.
- The alerting system was crude. For example, there was no visual cue to tell the user that a question had found new documents in the background. Also, the background searches did not exclude previously read documents.
- Headlines often ^{do not} don't give enough context. The headlines displayed in the question window were only about 60 characters long, making it difficult to identify which documents were useful without opening them. Furthermore, there was no provision to display the document's date or the name of the source it came from.

FIGS.
2,3,4
5

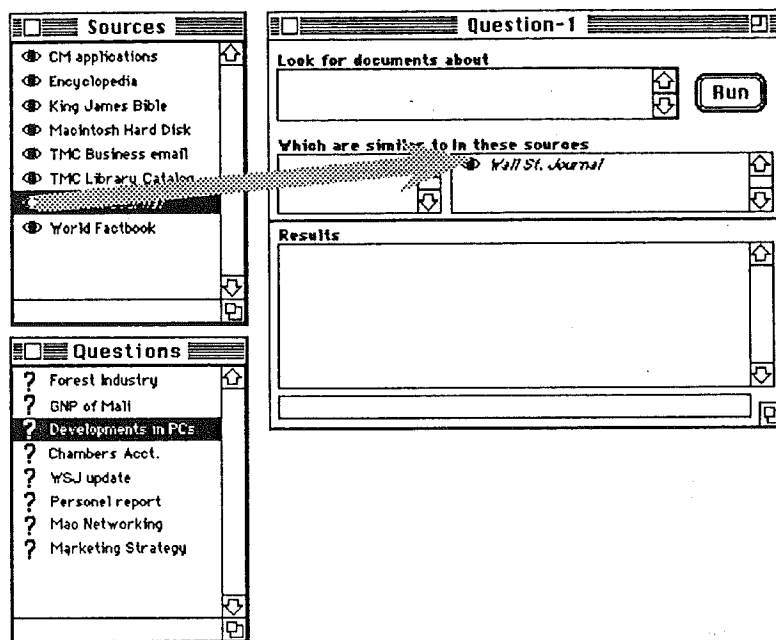


Figure 2

WAIStation's Sources and Questions windows store the user's personal objects. Dragging a source into a question window specifies that the question will contact the source in order to fulfill its charter.

JA-495

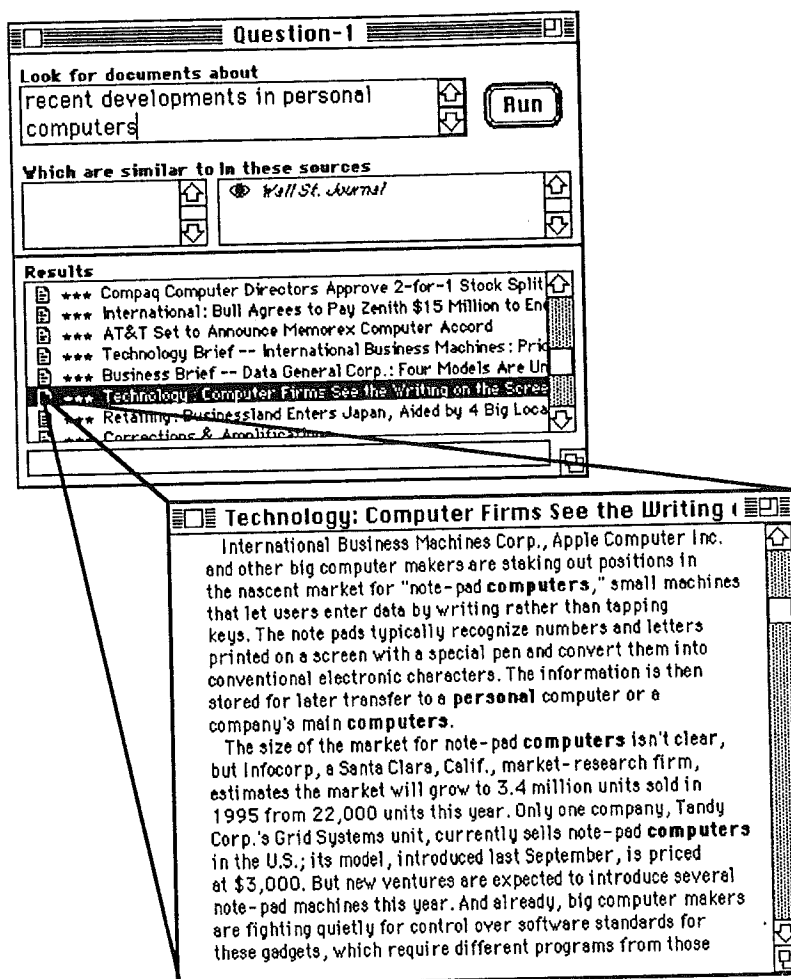


Figure 3

After running the question, results are displayed in a scrolling list. Double clicking on a result opens a document window. Query words are highlighted.

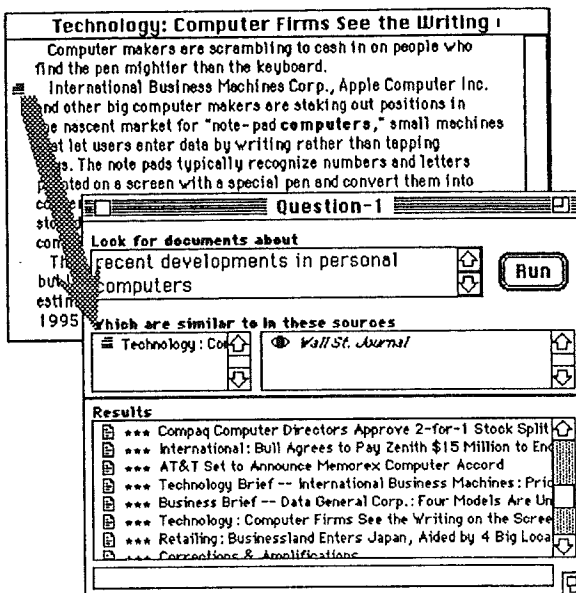
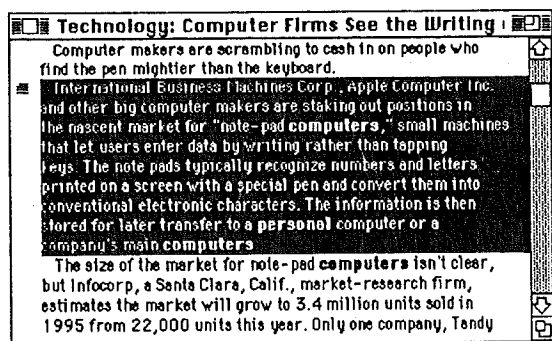


Figure 4 Relevance feedback is done by selecting a document or part of a document, and dragging the document or paragraph icon into a question.

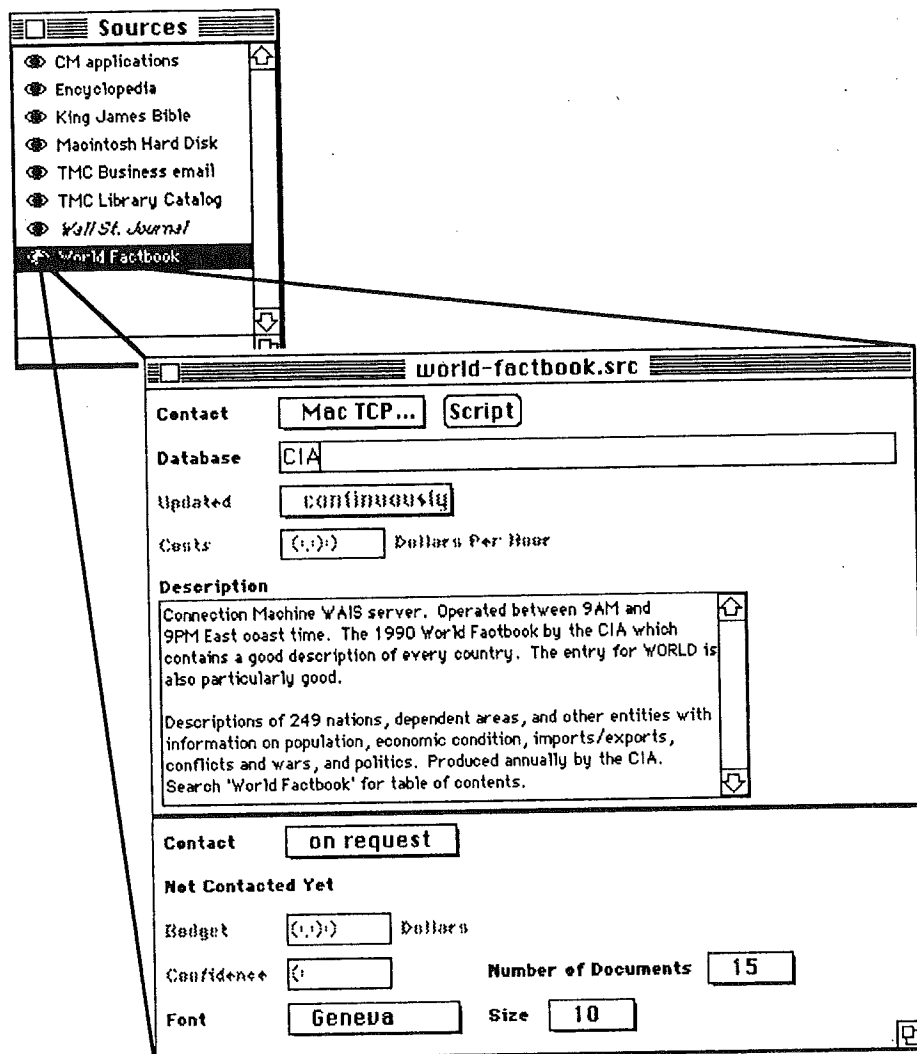


Figure 5 Double clicking on a source icon opens a source window.

X-WINDOWS BASED INTERFACE FOR WAIS: XWAIS

XWAIS At A Glance

Target Machine	X-windows terminals on <u>unix</u> machines
Effort	4 person-months
Number of Users	500
Status	Finished, freely distributed
Language	C
Communications	TCP/IP
Designer	Jonathan Goldman
Organization	Thinking Machines
Availability	Available anonymous FTP from /public/wais/wais*.tar.Z@think.com
Design Goals	Copy WAISStation so that we can leverage one design, portable, and based only on freeware Display data in many different formats (image, text, etc)
Used	Used in the Internet experiment. Heavy use by X users within Thinking Machines and outside
Problems	Installing it has caused many users to stumble. The number of variables (architectures, X directory structures) makes it difficult to make it portable, touch on the ability to handle different types (this is unique to this interface). Uses other programs to help (like interapplication communication)

*Ed: Note
Author has
been asked
to supply
single space
in paragraph.*

The WAIS interface for the X-Windows environment was developed for the Internet experiment to provide an X-Windows-based interface for a growing community. It was built to look as much like the Macintosh WAIS interface (WAISStation) as possible, given the limitations of the freely distributed X-Windows software. Since the metaphors in XWAIS are nearly the same as those for WAISStation, a user of one system can easily move to the other, without having to learn much ^{additional information} new. In fact, the underlying data structures are identical to those in WAISStation, so questions can be copied from a Macintosh to a UNIX machine running XWAIS, and used without modification.

XWAIS supports interactive WAIS access, including question entering, source selection, addition of relevant documents, and pieces of documents. Unlike WAISStation, XWAIS retrieves an entire document when requested, instead of just the parts being viewed. We decided this was acceptable, since the underlying networks for X will most likely be fast.

Since XWAIS runs under X-Windows, and was built for the UNIX operating system, it can take advantage of the tools available for these systems to display a wide range of document formats. A simple filter interface is provided in the application (as an X resource) to allow a user to select the tool required for a given type of document; ^{for example,} e.g. if the document is a Postscript file, xps can be used to view it. This is a feature that is not available in any of the other user interfaces described here.

MC In order to distribute this software without restriction, XWAIS uses the freely distributed Athena Widget set included in the X11R4 release from MIT. Although these widgets don't appear as attractive as some others that are available, they can be used to build a useful interface. Some aspects of this interface are restricted by the nature of the widgets available. XWAIS was built using the Xt X Toolkit Intrinsics, and allows a large amount of customization of the appearance of the display using X resources. The application relies heavily on the Xt resource mechanism, and will not run unless these resources are in place. The "object-oriented" feel of these widgets made building the interface rather easy, once the widget with the closest desired functionality was found. Finding the correct widget was the hardest part. Most of the actual behavior of the interface is controlled by "call-backs" ^{the} the methods that widgets inherit.

The XWAIS application is actually two separate applications: XWAIS, a simple shell for selecting sources and questions; and xwaisq, the application that actually performs WAIS transactions. The C code in xwaisq is also used in waisq, the shell-support program for GNU Emacs WAIS. This allows users to use simple UNIX facilities to submit questions created by xwaisq using waisq (e.g. a crontab entry to periodically query a server).

author should have been in all caps?

The implementation for XWAIS was done in C (6k lines) using the X11R4 release of X-Windows from MIT, the Xt X Toolkit Intrinsics, and the Athena Widget Set, included in the X-Windows release.

XWAIS is a text-based user interface built in a graphical window environment. Some additional graphical metaphors would be desirable, but the limited widget sets precluded that. It would take a considerably larger amount of work to add much graphics to this application. Perhaps some other X toolkit would provide simpler methods for doing this.

FIGS 6+7

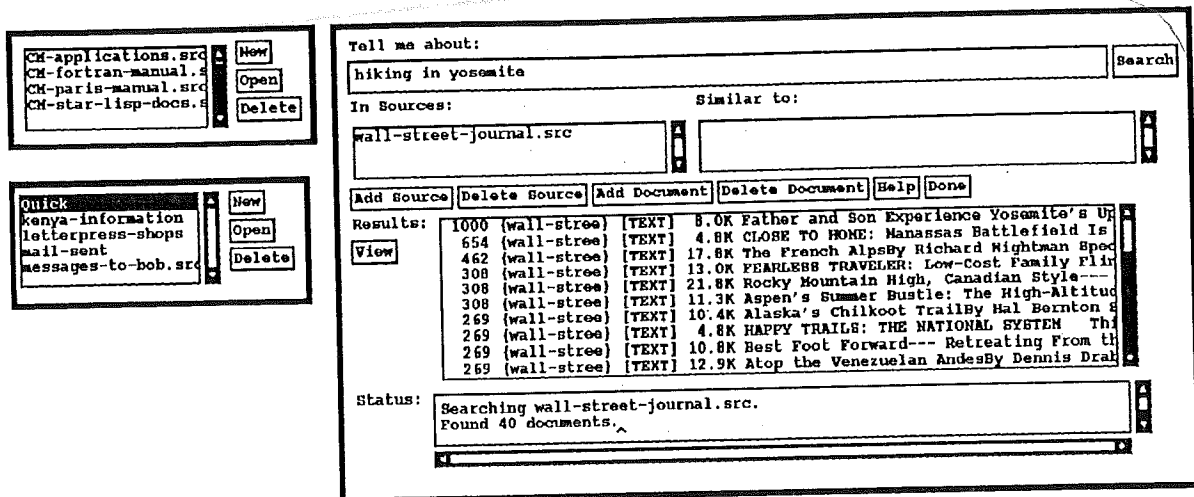


Figure 6 The XWAIS interface, including the Questions and Sources windows, and an open question.

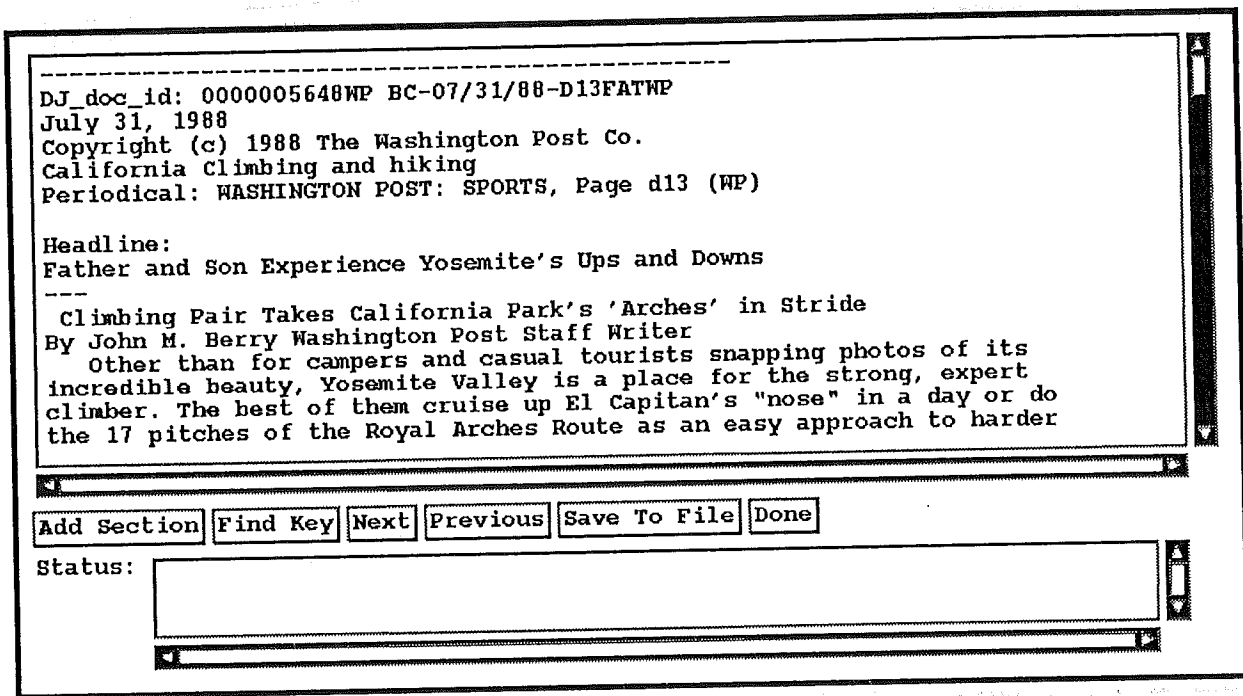


Figure 7 A document displayed in the XWAIS interface.

(12)

GNU EMACS WAIS INTERFACE: GWAIS

<i>GWAIS At A Glance</i>	
Target Machine	Terminals on <u>Unix</u> machines
Effort	2 person-months
Number of Users	500
Status	finished, freely distributed
Language	Gnu-Lisp, and C
Communications	TCP/IP
Designer	Jonathan Goldman
Organization	Thinking Machines
Availability	Available anonymous FTP from /public/wais/wais*.tar.Z@think.com
Design Goals	Copy WAISStation so that we can leverage one design, Use precedent from other <u>gnu-emacs</u> applications: RMAIL, dired
Used	Used in the Internet experiment with heavy use by some <u>gnu-emacs</u> users.
Problems	Dealing with the directory of servers. Using passive alerting.

B3
Yen
hp
box

Ed: I have requested paragraphs in single space from author

40 The WAIS interface on GNU-Emacs/Unix (GNU) was developed specifically for the Internet experiment for a technically strong user population. The reasons it was developed were: the large number of Emacs users, the extensibility, the ubiquitous nature of character display terminals, and the component nature of Emacs which meant WAIS could be integrated into email, boards, and programming tools.

40 The design of the interface was a cross between WAISStation and other emacs interfaces. The direct manipulation of WAISStation was replaced by command keys, as is common in emacs applications. The choice of command keys were modeled on the dired and RMAIL Emacs applications.

40 GWAIS allows users to access the interactive features of WAIS: question entering, relevance feedback, displaying document, and source selection. An extra feature, not found in the other interfaces, is an interface to an indexer for creating sources, but it appears that this feature is not heavily used. Furthermore it allows questions to be saved, but it depends on the user to automate the update of questions and sources using cron or other Unix tools. Graphic documents can be displayed on X-Windows terminals if the user has set up the environment variables.

E
E
E
author:
is this word correct?
not answered in my original query

The implementation of GWAIS was in Emacs (Lisp) (2K lines) and in C code (3K lines). About half of the time of a typical search and retrieval is spent in reading the data into Lisp.

author:
Should these be
LISP?
not change
as marks
in my
original
request

The screenshot shows the GWAIS for GNU-EMACS interface. At the top, the title bar reads "GWAIS for GNU-EMACS". Below it, the search query "chocolate cake" is entered. The interface displays a list of search results, organized into columns. The first column shows the document ID and size (e.g., 1000 2K). The second column shows the date (e.g., 01/10/90). The third column shows the document title (e.g., CHOC-CAKE-1(D)). The fourth column shows the source (e.g., USENET Cookbook). The fifth column shows the document type (e.g., CHOC-CA). The interface also includes a "Find Documents On" section with a dropdown menu set to "All" and a "On Sources" section with a dropdown menu set to "All". At the bottom, it says "Found 40 documents."

Find Documents On	On Sources	Similar To	Document ID	Date	Title	Source	Type
All	All	All	1000	2K	CHOC-CAKE-1(D)	USENET Cookbook	CHOC-CA
			725	3K	CHOC-CAKE-2(D)	USENET Cookbook	CHOC-CA
			607	2K	CHOKLADKAKA(B)	USENET Cookbook	CHOKLAD
			566	2K	CHOC-CAKE-4(D)	USENET Cookbook	CHOC-CA
			506	2K	CHOC-CAKE-5(D)	USENET Cookbook	CHOC-CA
			490	3K	CHOC-CAKE-3(D)	USENET Cookbook	CHOC-CA
			461	3K	CHOC-PUDDING-2(D)	USENET Cookbook	CHOC-PUDDI
			460	8K	CHOC-SLICES(D)	USENET Cookbook	CHOC-SL
			447	2K	SOURMILK-CAKE(D)	USENET Cookbook	SOURMILK-
			431	6K	PRINZ-TORTE(D)	USENET Cookbook	PRINZ-T
			420	2K	CHEESECAKE-4(D)	USENET Cookbook	CHEESECA
			420	2K	PUMPKIN-CAKE-2(D)	USENET Cookbook	PUMPKIN-CA
			416	2K	CHOC-PUDDING-1(D)	USENET Cookbook	CHOC-PUDDI
			413	2K	BLKFOREST-PIE(D)	USENET Cookbook	BLKFOREST
			412	2K	FUDGE-1(D)	USENET Cookbook	FUD
			412	1K	BUTTERNUTS(D)	USENET Cookbook	BUTTER
			403	3K	CHEESECAKE-6(D)	USENET Cookbook	CHEESECA
			401	2K	POUND-CAKE-2(C)	USENET Cookbook	POUND-CA
			399	1K	TRUFFLES-3(D)	USENET Cookbook	TRUFFL
			397	4K	ZUCCOTTO(D)	USENET Cookbook	ZUCC
			395	1K	CHEESECAKE-3(D)	USENET Cookbook	CHEESECA
			392	3K	CHEESECAKE-8(D)	USENET Cookbook	CHEESECA
			387	4K	CHEESECAKE-1(D)	USENET Cookbook	CHEESECA
			382	3K	CHOC-NUT-TORTE(D)	USENET Cookbook	CHOC-NUT-T
			382	2K	CHOC-CHIP-3(D)	USENET Cookbook	CHOC-CH
			380	3K	CARROTCAKE-1(D)	USENET Cookbook	CARROTCA
			379	2K	SABBATH-STEW(M)	USENET Cookbook	SABBATH-
			375	2K	CHOC-CHIP-2(D)	USENET Cookbook	CHOC-CH
			374	2K	CHOC-PIE-1(D)	USENET Cookbook	CHOC-P
			374	2K	CARROTCAKE-2(D)	USENET Cookbook	CARROTCA

Figure 8 The GWAIS interface, displaying the results of a relevance feedback search.

(H2)

SCREEN-BASED (TERMINAL) WAIS INTERFACE: SWAIS

SWAIS At A Glance

Target Machine	Terminals connected to <u>Unix</u> systems
Effort	1 person-month
Number of Users	900
Status	Beta
Language	C
Communications	TCP/IP
Designer	John Curran
Organization	NSF Network Service Center
Availability	To be included in WAIS release, anonymous FTP from /public/wais/wais*.tar.Z@think.com
Design Goals	Highly portable, provide straight-forward user interface, utilize existing application key mappings (rn, vi, Emacs), support multiple servers per query, allow for personal "source" directory and a common source directory, allow for useful source discovery via searches, provide simple active tool with little state (no question storage, relevance feedback, or passive notification).
Used	Internet users via Telnet: K-12 students, educators, user services staff, librarians, and (occasionally) network staff.
Problems	Dealing with the directory of servers. Lack of information in many server-returned records. Providing simple and uniform nomenclature. Planning for large numbers of sources.

B3
text
no
box

(as
(15)

To open WAIS to a wider community of users, an interface was developed to run on dumb terminals or over Telnet sessions. It is called "SWAIS" for Screen WAIS since it uses a character display terminal screen for the interface. The user communities that this interface can serve are dial-in users, Telnet users, and low-end terminal users.

The design of the interface involved three screens: a single screen listing all known servers that the user could pick from; a list of search result documents headlines; and a document display screen. Listing all servers and

allowing users to select which servers to use encourages users to ask questions of multiple servers. Unlike the other interfaces, the sources list shows what site runs it and how much it costs (if anything). The resulting document screen includes headlines and number of lines, but its innovation is to show the source. 7

It does not handle relevance feedback or downloading new sources from the directory of servers. Another drawback is using it with large numbers of sources, since moving around the list requires scrolling. On the other hand, this server has proven to be very popular on the Internet. Because of its ease of use, all a user has to do is use Telnet to a specific machine to use it.

FIGS
9, 10,
11

Ans:
Clear
?

SWAIS			
Source Selection		Sources: 79	
#	Server	Source	Cost
55:	[cans.think.com]	potent-sampler	Free
56:	[microworld.media.mit]	poetry	Free
57:	[lambada.oit.unc.edu]	rec.cook	Free
58:	[lambada.oit.unc.edu]	rec.pets	Free
59:	[cans.think.com]	risks-digest	Free
60:	[pit-manager.mit.edu]	rkba	Free
61:	[quake.think.com]	sample-books	Free
62:	[quake.think.com]	sample-pictures	Free
63:	[129.71.11.2]	sonnet-ada-archives	Free
64:	[cans.think.com]	sun-spots	Free
65:	[talon.ucs.onst.edu]	supreme-ct	Free
66:	[quake.think.com]	tao-library	Free
67:	[uncx1.oit.unc.edu]	unc-jobs	Free
68:	[ariel.its.unimelb.ED]	unimelb-research	Free
69:	[quake.think.com]	unix-manual	Free
70:	[next2.oit.unc.edu]	unix.FAQ	Free
71:	[violet.cs.uq.oz.au]	usenet-Faq	Free
72:	[cans.think.com]	usenet-cookbook	Free

Keywords: What Yeats poem is about a falcon and falconer?

<space> selects, w for keywords, arrows move, <return> searches, q quits, or ?

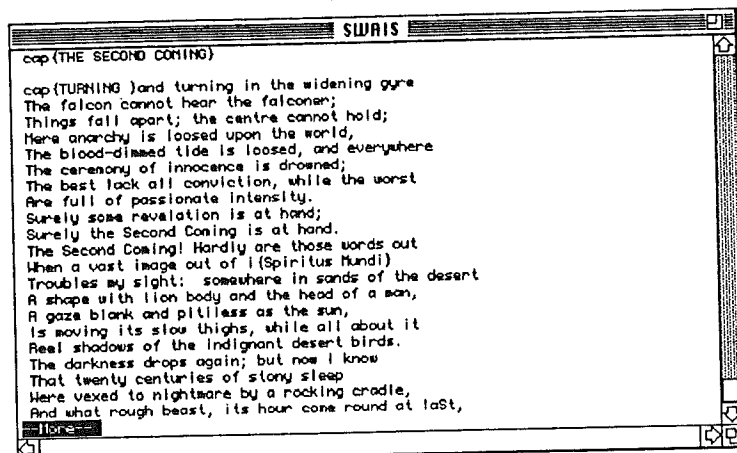
Figure 9

The SWAIS query building screen. The poetry source is selected, and search terms are entered. This interface does not currently support relevance feedback.

SWAIS	
Search Results Help	
J, ^M	Move Down one item
K, ^P	Move Up one item
##	Position to item number ##
<space>	Display current item
<return>	Display current item
	Pipe current item into a unix command
U	View current item information
S	Specify new sources to search
u	Use it; add it to the list of sources
w	Make another search with new keywords
h	Show this help display
H	Display program history
q	Leave this program

Press any key to continue

Figure 10 The SWAIS help screen.



(C) Figure 11 A document displayed in SWAIS.

(H2) **THE ROSEBUD INTERFACE: REPORTERS AND NEWSPAPERS ON THE MACINTOSH**

Rosebud At A Glance

Target Machine	Macintosh II, color screen
Number of Users	25
Status	Finished; internal use
Language	Smalltalk, MPW-C
Communications	TCP/IP using IPC package
Designers	Charlie Bedard, David Casseres, Steve Cisler, Tom Erickson, Ruth Ritter, Eric Roth, Gitta Salomon, Kevin Tiene, Janet Vratny-Watts.
Organization	Apple Computer
Availability	Only internally to Apple ATG
Design Goals	Serve as research platform for interface and architectural explorations. Allow ordinary users to create personalized information flows; support passive alerting, scanning, and capture of information.
Used	Used in various internal tests; not available for the Internet experiment.
Problems	No good interface mechanisms for providing users with convenient access to large numbers of servers.

Ed:
The requested
single page
in paras
from author.

41 Rosebud is a project within Apple Computer's Advanced Technology Group. Its principle objective is to serve as a platform for investigations into what is needed to make remote information accessible and useful to ordinary Macintosh users. The investigations have two foci: human interface components and techniques, and system architecture issues. In this article we focus exclusively on the human interface aspects of Rosebud.

41 The Rosebud Server System is similar to the WAIS system in that it uses the Z39.50 protocol to access multiple, remote databases; it differs from them in that it contains extra underpinnings for making information access an integral part of the Macintosh environment. Specifically, the Rosebud Server System allows users to create autonomous, ongoing "agent" processes which access, update, and present information from local and

remote sources. The Rosebud system does not currently provide access to the Internet WAIS servers (for reasons of network security, rather than basic incompatibilities), and is not publicly available.

Design Rationale

The design of the Rosebud interface began with a study of the practices and problems of ordinary information users. The principle focus was on information users at KPMG Peat Marwick in San Jose, the original client site for WAIS; in addition, several groups of users of online information services within Apple were also studied (Erickson & Salomon, 1991). Interviews with accountants at Peat Marwick enabled the designers to put together a schematic of how information (mostly paper-based information) flowed through their offices (Figure 12).

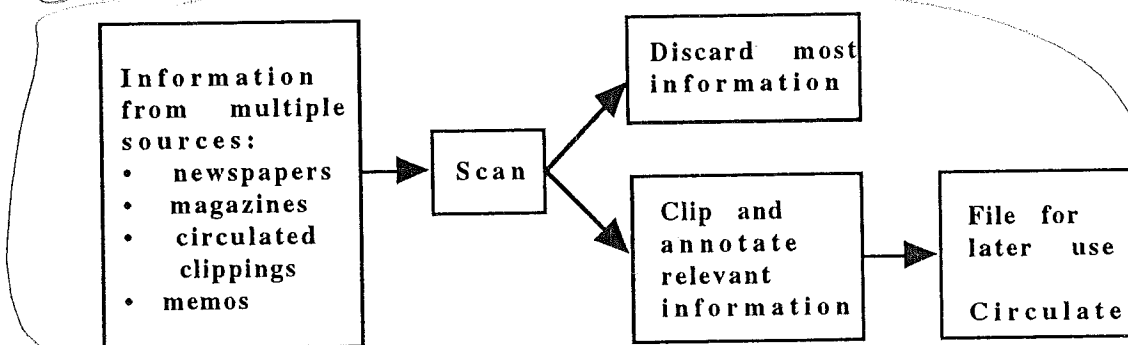


Figure 12. Information flow through accountants' offices.

Several features of this schematic informed the design of Rosebud. First, information typically came to the accountants via newspapers, magazines, and memos; instances where the accountants went out of their way to search for information were less frequent. Second, the accountants never talked about "reading" information; they always spoke of scanning, or skimming it ^{if} they didn't ^{not} have time to read it. This suggested that a good interface should provide a way for the users to scan retrieved information quickly. Third, accountants remarked that they discarded most information, including information that might be useful. Potentially useful information was discarded for two reasons: the accountants did not have the physical space to store everything, and they knew from experience that if they tried to save too much, they would not be able to find anything later when they actually needed it. This suggested that giving users access to remote information was just half the problem; users also needed tools for archiving, organizing, and retrieving information. Finally, when users did come across information that seemed worth saving, they typically would cut it out (the accountants used, almost exclusively, paper-based information), and then they would annotate it by circling, underlining, or jotting a few notes in the margin. Annotation turned out to be an important concept: not only did it help the

user who annotated when the information was retrieved later, but it also helped others scan the information more quickly when copies were passed on to them.

The consequence of these observations was a design for a system which allowed users to define topics of interest which would be retrieved automatically, and would then permit them to scan those items and save them into an environment where they could be annotated, organized, and retrieved.

Human Interface

The Rosebud interface design has three components: reporters, newspapers, and notebooks. Reporters are for retrieving information. Users give reporters assignments which specify what to look for, and where to look. This is shown in figure 13: users enter words describing the information in which they are interested, check off the information sources they wish the reporter to search, and, if they so choose, automate the reporter so that it searches the databases on a daily or weekly basis. Upon pressing the "Search" button in the assignment window, a reporter is created, performs the search, and returns with a list of results (Figure 14).

Figure 13 Creating a reporter in the assignment window.

The reporter window (Figure 14) provides users with a variety of ways to look over their results, and refine their queries. The results are shown in the "Best Guesses" pane. (The name "Best Guesses" was chosen to provide some indication that inaccuracy could be expected; our observations of users had shown that they were often mystified by some of the items that showed up as the results of searches.) The asterisks to the left of items indicate their relative relevance, and the pop-up menu above the pane allows users to order the list by date or relevance. Simply selecting an item shows a preview of it—a short excerpt with search terms highlighted in color and boldface (Figure 15). Previews are useful because users can get a look at a small part of the item without incurring the overhead of downloading the whole article over the network. Users also have

the options of saving articles to their disks or opening them for viewing. Finally, having looked over their results, users can refine their search in the bottom pane of the window.

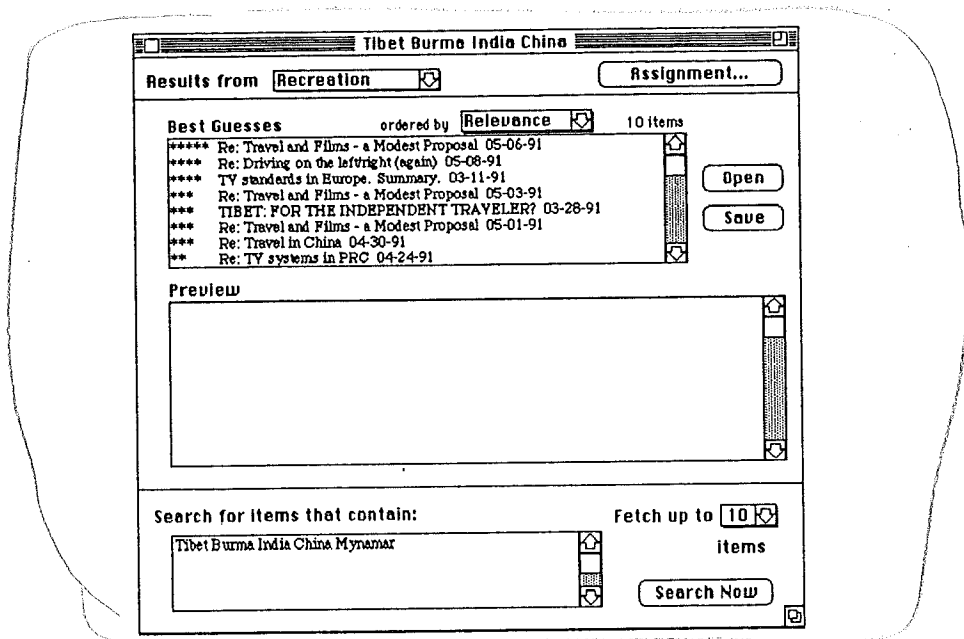


Figure 14 The reporter window contains the results of the search and provides means for previewing, opening, and saving results.

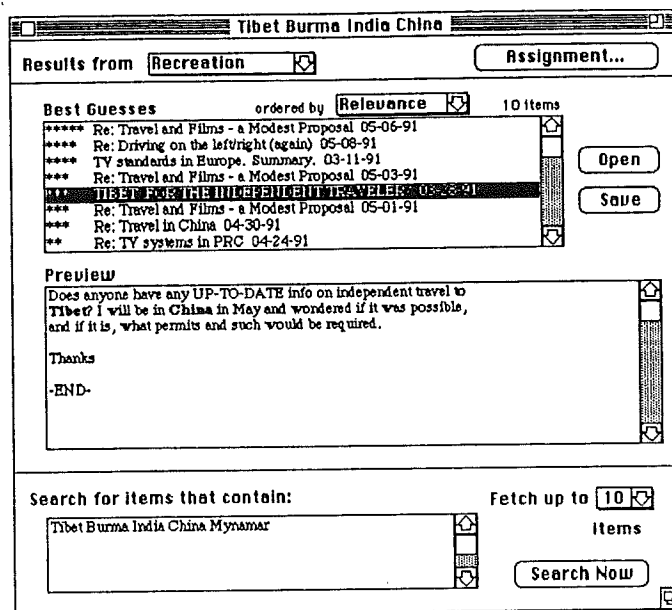


Figure 15 The reporter window makes it easy to scan through hits. Clicking on a retrieved item generates a preview which shows an excerpt to the hit with the search terms (Tibet and China) highlighted in boldface. The user can refine the query in the lower pane of the window.

The above sequence occurs whenever a user creates a new reporter. However, since users are likely to use many reporters, and because the initial user studies indicated that ways of skimming through incoming information were important to the accountants, the newspaper was provided to support rapid scanning of new information. The model of a newspaper is quite simple (Figure 16): on the left is an index column which contains the names of all reporters, and to the right are two columns of news. Each reporter "owns" one news column and publishes the title, date, and an excerpt of each item in its column. The columns scroll independently, using "minimalist" scroll bars to prevent the multiple scroll bars from visually overloading the screen. If an excerpt seems interesting, double clicking on it opens the full article in a window, from which it can be viewed, printed, or saved. Thus, rather than having to open up a dozen reporters every morning to see what's new, the user can go to one place, the newspaper.

FIG 16

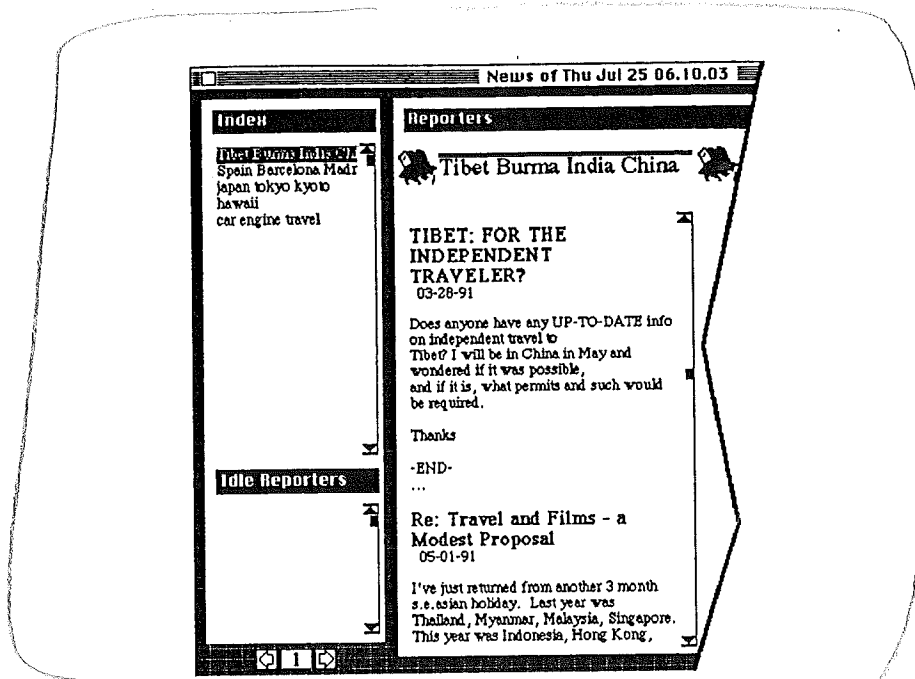


Figure 16 The newspaper allows users to quickly scan through new items retrieved by the reporters which are working automatically.

The newspaper can also serve as a control center for the Rosebud interface. The user can open a reporter by clicking on its name or icon at the top of its news column. Consequently, if a reporter's column has strayed from the desired topic, the user can quickly get to the reporter and revise its assignment. The index also lists inactive reporters (those either not automated, or that have not found anything new since the last newspaper), so they, ^{also} too, can be opened, and automated or otherwise adjusted.

A third component of the Rosebud interface ^{the notebook} was designed but not implemented. Notebooks are environments within which users may save, annotate, and organize retrieved information. Notebooks were designed in response to the observations of Peat Marwick accountants, which indicated the need for an environment which supported the way accountants worked ^{in particular}, notebooks were intended to support annotation, and re-finding retrieved information at a later date. A particularly nice feature of the notebook design was its use of annotations as landmarks for re-finding information. The notebook design, and its rationale, is described in ^{and} Erickson & Salomon, (1991).

Implementation

The Rosebud system consists of six parts: (1) a human interface application written in SmallTalk/V (to facilitate the rapid changes in the interface necessary to effectively conduct interface design research); (2) a search manager package which implements the autonomous agent functionality and formulates Z39.50 queries

for (3) remote Z39.50 servers implemented in MPW C that automatically index items placed in their input folders by (4) HyperCard stacks that download new items from a Net News server; (5) a file manager component (MPW C) that does all of the file I/O and compaction for reporters and newspapers; and (6) directory servers which allow the various components to find one another. All of these components are written as separate applications and communicate with one another using a prototype IPC that runs over TCP/IP. The file manager and search manager applications run in the background under MultiFinder, enabling Rosebud to access information and construct newspapers while the human interface application is not running. Like the other WAIS interfaces, Rosebud uses the WAIS protocol package. The human interface was designed for Macintosh II class machines, with 13-inch color screens.

H2 Observations and Testing Results

The Rosebud human interface was subjected to informal testing on 14 users. Users were told only that Rosebud was an application for finding information, and then given a particular topic to find information on. They were given no help or documentation. Note that, although informal, this type of testing is very stringent, in that users approach the application knowing almost nothing about what it is, or why they would actually use it. Data collection consisted simply of recording their questions, observations, and problems as they went along, administering a post-test questionnaire, and then asking them a few, open-ended questions. Here are a few of the more general observations.

- Over 80% of those who tried the Rosebud interface responded very positively to it, and said that they would use something with its capacities as part of their daily work routine. Two thirds of users indicated that they would usually use newspapers to browse through information (instead of reporters).
- At the end of the test, over two thirds of the users said they liked the metaphors of reporters and newspapers; however, almost all users had some difficulty in getting started. The typical problem was that users did not associate reporters with a way of retrieving information. When asked to find information, users first looked for an item called "search"; when they didn't find this, they usually turned to the newspaper, which is, in fact, where they look for information on a daily basis. It is possible that this problem can be remedied by minor interface changes (e.g. putting a "New Reporter" item in a search menu); alternatively, it may be that the metaphor is inappropriate.

B3

- ④ • A number of users were led astray because they had conceptual models of information retrieval based on their familiarity with query languages and structured databases. Such users tended to be wary of entering search terms because they were not sure of the appropriate syntax, and did not understand what "relevance" meant. Those who did know the meaning of relevance wanted to know how the information server calculated it.
- ④ • Users liked previews especially the feature of highlighting keywords in boldface. They wanted to see boldface keywords in the newspaper and article windows. Users also wanted the ability to select text in the newspaper and article windows and change the style or font themselves, so that they could annotate significant items. This parallels practices observed in our initial observations of accountants, where we found that annotation plays several important roles.
- ④ • A variety of low-level interface problems, due to terminology or graphic design were discovered. Some examples: users did not usually recognize the asterisks in the "Best Guesses" window as indicators of relevance; users didn't think that "idle reporters" was a good name, and said that it was very important to distinguish between reporters that had found nothing, and those that weren't looking.

The testing described above focused on how usable Rosebud was with users' first exposure. In the next phase of testing, a small set of users will be observed over the course of a month, in which they have the option of using Rosebud from their desktop machines to access meaningful data. This phase of testing will allow a more realistic assessment of Rosebud, in that it will last long enough to permit users to build up their own set of reporters, and to access newspapers that contain information of personal import.

CONCLUSION

- ④ This article has described five interfaces developed to provide access to distributed systems of information servers. The interfaces presented here were developed with different constraints in mind, so it is not useful to compare them directly; instead, they may serve as examples of differing responses to issues such as screen size, workstation power and intelligence, communication speeds, and user needs and practices.
- ④ The interfaces designed so far have addressed some of the critical issues for end-users to accomplish interactive searches in a wide-area network. These include ways of finding which information servers contain relevant information; supporting searching by ordinary users; and supporting browsing of, and passive alerting about, newly retrieved information. The alerting aspects of the interfaces have not been tested much in this

LS:

Appendix

day(?)

environment due to the lack of appropriate data sources for this type of searching. It is probably fair to say that any of the design solutions described here can be improved upon by further work.

9) The WAIS Internet experiment has revealed a number of issues requiring further work. In the Internet environment we have observed (in the logs of user queries) that users have a difficult time finding out what is in a database, thus demonstrating that there is a lack of browsing or scanning facilities in the interfaces, protocol, and servers, as well as a general shortage of descriptive information about databases.

9) Finally, a variety of other issues were raised during the studies of the Peat Marwick accountants who have received little or no work. Document layout is one such problem. Accountants mentioned that sometimes they want to retrieve documents not because of the information they contain, but to look at their layouts (accountants will often examine successful proposals to a client when preparing a new proposal). More generally, users regard pictures, diagrams, tables, and charts as essential components of a document's content. Unfortunately, support for different document formats, and for the retrieval and display of non-textual information within them is very limited on most existing clients.

9) Another issue is called the boilerplate problem. Accounting documents often contain a large amount of boilerplate ¹/_m standard text which varies little from document to document. What tools are needed to allow users effectively to retrieve, order, and browse a large set of documents which are 95% similar? Note that boilerplate is characteristic of a wide variety of business proposals and legal documents, not just accounting documents. In fact, the analog to boilerplate occurs in scientific documents in which standard terms and descriptions are used to describe procedures and methods used in an investigation.

9) A number of other issues remain to be addressed. Users are very interested in being able to see what queries other users are conducting, and what information servers and articles are most popular. A frequent suggestion is to allow users to rate the "goodness" of articles they retrieve. However, in a commercial setting, information about the kind of questions being posed by a particular company or person can be revealing and valuable. Clearly, the utility that such information could provide must be balanced by concerns about confidentiality and privacy, and mechanisms for user control of descriptive information are essential. Other issues include how to control the pricing, copyright, and distribution issues which accompany "for-pay" information.

9) In summary, there is an immense amount of work to be done. A central part of this work involves further research and development of interfaces. We have made the WAIS system publically available in the hope that designers will find that it ¹/_m with its common protocol and defined infrastructure ¹/_m can serve as a platform from which to pursue these, and other, research issues.

11) Appendix

FOR MORE INFORMATION ON THE WAIS SYSTEM

COMP:
Acknowledgments
were from msp 35

The success of a distributed system of information servers depends on a critical mass of users and information services. To encourage development and use, Thinking Machines is making the source code for a WAIS protocol implementation freely available. While this software is available at no cost, it comes with no support. We hope that it will facilitate others in developing servers and clients.

For more information, please contact:

Barbara Lincoln Brooks (barbara@wais.com) WAIS Inc. 1040 Noel Drive Menlo Park, CA 94025

REFERENCES

Davis, F., Kahle, B., Morris, H., Salem, J., Shen, T., Wang, R., Sui, J., & Grinbaum, M. (1990) WAIS Interface Protocol/Functional Specification, Thinking Machines Corporation, April. Available via anonymous ftp: /pub/wais/doc/wais-concepts.txt@quake.think.com or wais server wais-docs.src

Dongarra, J., & Grosse, E. (1987) Distribution of Mathematical Software Via Electronic Mail, CACM, Vol. 30, pp. 403-407.

Egan, D., Remde, J., Gomez, L., Landauer, T., Eberhardt, J., & Lochbaum, C. (1989) Formative Design - Evaluation of SuperBook, ACM Transactions on Office Information Systems, January.

Erickson, T., & Salomon, G. (1991) Designing a Desktop Information System: Observations and Issues, In Proceedings of the ACM Human Computer Interaction Conference, ACM Press, April, pp. 49-54.

Ginther-Webster, K. (1990) Project Mercury, AI Magazine, pp. 25-26.

For information on the Free Software Foundation and the GNU project, see: /pub/gnu/*@prep.ai.mit.edu. For emacs see: /pub/gnu/emacs*.tar.Z@prep.ai.mit.edu. Free Software Foundation, 675 Massachusetts Avenue, Cambridge, MA 02139.

Kahle, B., & Medlar, A. (1991a) An Information System for Corporate Users: Wide Area Information Servers, April. Online Magazine, September.

Kahle, B., Goldman, J., Morris, H., & Shen, T. (1991b) Electronic Publishing Experiment on the Internet: Wide Area Information Servers. In preparation.

Malone, T., Grant, K., & Turback, F. (1986) The Information Lens: An Intelligent System for Information Sharing in Organizations; In Human Factors in Computing Systems, CHI'86 Conference Proceedings, Boston, MA, ACM, New York; pp. 1-8.

For more information on the Mosis fabrication facility write to mosis@mosis.edu.

NeXT Computer Inc., 900 Chesapeake, Redwood City, California 94063 (415) 366-0900.

Z39.50-1988: Information Retrieval Service Definition and Protocol Specification for Library Applications, National Information Standards Organization (Z39), P.O. Box 1056, Bethesda, MD 20817 (301) 975-2814. Available from Document Center, Belmont, CA Telephone (415) 591-7600.

ON Technology Inc., 155 Second Street, Cambridge, MA 02141 (617) 876-0900.

Verity Inc., 1550 Plymouth Street, Mountain View, CA 94043 (415) 960-7600.

Author: please supply page nos. + vol.

Am: City: OKAY

E

Author: which is it? please give page nos.

Author: please give page nos. Journal or book?

Copy: drop this heading a line or two to separate it from the body

H1

ACKNOWLEDGMENTS

At Thinking Machines: Ottavia Bassetti, Franklin Davis, Patrick Bray, Danny Hillis, Rob Jones, Barbara Lincoln, Gordon Linoff, Chris Madsen, Gary Rancourt, Sandy Raymond, Tracy Shen, Craig Stanfill, Steve Schwartz, Robert Thau, Ephraim Vishniac, David Waltz, and Uri Wilensky.

At the National Science Foundation: Suzi, Alex, et al.

Place Figs. 1-11
in text near
where Ed. marked
them.

Write query about Citing Figures 1-11.



605 Third Avenue
New York, New York 10158
(212) 850-6000
Telex 12-7063
Cable JONWILE
New York, Chichester,
Brisbane, Toronto

John Wiley & Sons, Inc. Publishers

Prod. No. _____

Dear Author:

Under U.S. copyright law the transfer of copyright from the author should be explicitly stated in writing to enable the publisher to publish and disseminate the author's work to the fullest extent. The following agreement must be signed and returned to the editorial office of the journal before we can process your manuscript. Please note that if the manuscript was prepared as a work made for hire, this agreement must be signed by the employer, as author.

Publication Agreement

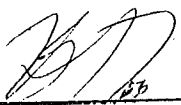
The undersigned author has submitted a manuscript entitled "Interfaces for Distributed Systems of Information Servers" for publication in the _____ published by John Wiley & Sons, Inc.

- A. The author transfers to John Wiley & Sons, Inc. the exclusive rights comprised in the copyright of the article, if the article is accepted for publication, except that the author retains the following:
1. All proprietary rights, other than copyright, such as patent rights.
 2. The right to make copies of all or part of the work for the author's use in classroom teaching.
 3. The right to use, after publication, all or part of this material in a book by the author.
 4. The right to make copies of the work for internal distribution within the institution which employs the author.
 5. The right to use figures and tables from the work, and up to 250 words of text, for any purpose.
 6. The right to make oral presentation of the material in any forum.
 7. The right to veto or approve permission granted by the publishers to a third party to republish all or a substantial part of the article.

In the case of a work prepared under U.S. Government contract, the U.S. Government may reproduce, royalty-free, all or portions of the article and may authorize others to do so, for official U.S. Government purposes only, if the contract so requires.

The author agrees that all copies made under any of the above conditions will include notice of the copyright to the publisher.

- B. The author warrants that the manuscript is the author's original work. If the work was prepared jointly the author agrees to inform co-authors of the terms of this agreement. It is submitted only to this journal, and has not been published before. (If excerpts from copyrighted works are included, the author will obtain written permission from the copyright owners and show credit to the sources in the manuscript.) The author also warrants that the article contains no libelous or unlawful statements, and does not infringe on the rights of others.

 5-19-93
Author's signature and date

Check one:

☐ Employer; work made for hire

☐ Work of the U.S. Government

☒ Author's own work

Brewster Kahle
Typed or printed name

WAIS Inc.
Institution or company

Note to Government Employees

A work prepared by a government employee as part of his/her official duties is called a "work of the U.S. Government," and is not copyrightable. If it is not part of the employee's official duties it may be copyrighted. If this is a work of the U.S. Government, the author may cross out Section A of this agreement and attach a copy of applicable policy of author's agency.

If the work was prepared jointly, and any co-author is not a U.S. Government employee, that author must be delegated by the co-authors to sign the *complete* agreement.

Jonathan Goldman's Bio
As of 4-13-93

Jonathan Goldman is a Software Engineer living in Menlo Park California.

After receiving a B.S in Chemical Engineering from the University of Maryland in 1982, Mr. Goldman worked for a radiation chemistry laboratory at the University. It was there that he became interested in computers, working with the latest Hewlett Packard desktop systems. He then worked for a small defense contractor, which led to the Perkin Elmer Advanced Development Center, where he used one of the first Connection Machine systems from Thinking Machines Corporation.

Mr. Goldman is currently employed by Thinking Machines Corporations developing advanced database systems. He continues to pay close attention to developments in the area of networked information systems.

Tom Erickson's Bio
as of 3-15-93

Tom Erickson is an interface designer, analyst, and researcher with Apple Computer's Advanced Technology Group where his responsibilities include designing interfaces for future technologies and applications. Tom has been involved in user interface design since 1979, and has worked as a cognitive psychologist, a programmer, and a writer, in addition to his design activities.

John Curran's Bio

John Curran is Network Analyst with Bolt Beranek and Newman, Inc. of Cambridge, Massachusetts. Mr. Curran works on a variety of projects including the National Science Foundation Network Service Center (NNSC) and the New England Academic and Research Network (NEARnet). John also participates in the Internet Engineering Task Force (IETF) in the areas of operational requirements, user services, and network protocol design.

will call me
3/25.

cm of 6015 4/5
4/5 Diane circa 2/12-
850-16660
muted agreement
to Diane

The undersigned has submitted a manuscript entitled "Interfaces of Distributed Systems of Information Servers" for publication in Perspectives on Electronic Libraries published by John Wiley & Sons, Inc.

- A. Brewster Kahle transfers to John Wiley & Sons, Inc. the non-exclusive right to print, publish, sell and license the manuscript.

Brewster Kahle retains all other rights in the manuscript including, but not limited to, the right to publish the manuscript elsewhere after publication of Perspectives on Electronic Libraries, and the right to distribute the manuscript electronically as a technical report.

John Wiley & Sons, Inc. agrees to publish a notice of copyright in the manuscript in the name of Brewster Kahle.

- B. The author warrants that the manuscript is the author's original work. If the work was prepared jointly the author agrees to inform co-author's of the terms of this agreement. (If excerpts from copyrighted works are included, the author will obtain written permission from the copyright owners and show credit to the sources in the manuscript.) The author also warrants that the article contains no libelous or unlawful statements, and does not infringe on the right of others.



4-5-93

Author's Signature and date

Brewster Kahle

Typed or printed name

WAIS Inc.

Institution or company

Check one:

___ Employer; work made for hire

___ Work of the U.S. Government

☒ Author's own work

The undersigned has submitted a manuscript entitled [REDACTED] for publication in [REDACTED] published by John Wiley & Sons, Inc.

- A. [REDACTED] transfers to John Wiley & Sons, Inc. the non-exclusive right to print, publish, sell and license the manuscript.

[REDACTED] retains all other rights in the manuscript including, but not limited to, the right to publish the manuscript elsewhere after publication of [REDACTED] the right to distribute the manuscript electronically as a [REDACTED] technical report.

John Wiley & Sons, Inc. agrees to publish a notice of copyright in the manuscript in the name of [REDACTED]

- B. The author warrants that the manuscript is the author's original work. If the work was prepared jointly the author agrees to inform co-authors of the terms of this agreement. It is submitted only to this journal and has not been published before. (If excerpts from copyrighted works are included, the author will obtain written permission from the copyright owners and show credit to the sources in the manuscript.) The author also warrants that the article contains no libelous or unlawful statements, and does not infringe on the rights of others.

[REDACTED]
Author's signature and date

2/4/93

Check one:

☐ Employer; work made for hire

☐ Work of the U.S. Government

☐ Author's own work

[REDACTED]
Typed or printed name

[REDACTED]
Institution or company

JD - Inform of
Should get fairly soon
to run parser

6/19-7/27-9/20
Exham
6mgs file
progr. counts

Diane Ceria
JW + Sons, Inc.
605 3rd Ave
NY, NY, 10158
212-850-6660